

**WHAT IS CLAIMED IS:**

1           1. A method of generating object lifetime statistics based on run-time  
 2 observations, the method comprising:  
 3           selecting from amongst object instances of an observed category, a sampled  
 4           subset of the object instances allocated in one or more execution  
 5           threads of a computational system;  
 6           coincident with allocation of a sampled instance of an object, establishing a  
 7           weak reference thereto and associating therewith information  
 8           indicative of at least allocation time; and  
 9           referencing the sampled instances at run-time via the weak references and  
 10           updating the object lifetime statistics based on the associated allocation  
 11           time and then-current state.

1           2. The method of claim 1,  
 2           wherein the then-current state includes garbage collection state of sampled  
 3           instances.

1           3. The method of claim 1,  
 2           wherein the computational system includes a garbage collector; and  
 3           wherein the object lifetime statistics updating is performed in response to a  
 4           determination by the garbage collector that one or more sampled  
 5           instances have become unreachable.

1           4. The method of claim 1,  
 2           wherein the computational system includes a generational garbage collector;  
 3           and  
 4           wherein the object lifetime statistics updating is performed in response to a  
 5           determination by the generational garbage collector that one or more  
 6           sampled instances have become unreachable or have been promoted  
 7           from a younger generation to an older generation.

1           5. The method of claim 1,

2 wherein the object lifetime statistics updating is performed by periodically  
3 accessing the sampled instances.

1 6. The method of claim 1,  
2 wherein the object lifetime statistics updating is performed by purging a subset  
3 of the object lifetime statistics.

1 7. The method of claim 1,  
2 wherein the object lifetime statistics are represented as histogram of lifetimes.

1 8. The method of claim 1,  
2 wherein the object lifetime statistics are represented mean lifetimes.

1 9. The method of claim 1,  
2 wherein the object lifetime statistics are calculating using an average birth  
3 date.

1 10. The method of claim 1,  
2 wherein an observed category corresponds to an object class.

1 11. The method of claim 1,  
2 wherein an observed category corresponds to a garbage collection generation.

1 12. The method of claim 1,  
2 wherein the associated information indicative of at least allocation time is  
3 further indicative of allocation site.

1 13. The method of claim 1,  
2 wherein the associated information indicative of at least allocation time is  
3 further indicative of an allocating one of the execution threads.

1 14. The method of claim 1,

2 wherein the associated information indicative of allocation time is encoded as  
3 one or more of allocation count, system time, CPU time, byte count,  
4 and garbage collection count.

1 15. The method of claim 1,  
2 wherein the weak reference is of a type not considered in reachability analysis  
3 of a garbage collector.

1 16. In an automatically reclaimed storage environment, a method of sampling  
2 instances of software objects during respective lifetimes thereof, the method  
3 comprising:

4 establishing weak references to respective of the sampled instances, each of  
5 the weak references identifying at least one respective sampled  
6 instance;  
7 associating allocation-time information with each sampled instance; and  
8 accessing the sampled instances via the weak references and performing an  
9 action based at least in part on a state of one or more of the sampled  
10 instances and respective allocation-time information.

1 17. The method of claim 16,  
2 wherein the weak reference establishing includes storing in a data structure a  
3 reference not considered in reachability analysis of the automatically  
4 reclaimed storage environment.

1 18. The method of claim 16,  
2 wherein the sampled instances include a representative subset of a category of  
3 software objects.

1 19. The method of claim 18,  
2 wherein the category is object class specific.

1 20. The method of claim 18,  
2 wherein the category is call-site specific.

1 21. The method of claim 18,  
2 wherein the category corresponds to an activation record stack profile.

1 22. The method of claim 18,  
2 wherein the category covers an abstract class or interface.

1 23. The method of claim 18,  
2 wherein the category is specific to a particular garbage collection space.

1 24. The method of claim 16, wherein the allocation-time information includes  
2 one or more of:  
3 time of allocation;  
4 allocation site;  
5 allocating thread; and  
6 object type.

1 25. The method of claim 16, further comprising:  
2 selecting at allocation time the sampled instances from amongst all instances  
3 of a particular type.

1 26. The method of claim 25,  
2 wherein the selecting is based on allocation buffer overflow.

1 27. The method of claim 25,  
2 wherein the selecting is based on a subset of allocations for each type of  
3 sampled software object.

1 28. The method of claim 27,  
2 wherein the subset includes a pseudo random distribution of the allocations.

1 29. The method of claim 27,  
2 wherein the subset includes a deterministic distribution of the allocations.

1 30. An object sampling facility for a computational system, the object  
2 sampling facility comprising:  
3 a weak reference construct implemented by the computational system; and  
4 an object fingerprinter responsive to a storage allocator of the computational  
5 system, the object fingerprinter associating (1) allocation-time  
6 information and (2) an instance of the weak reference construct with at  
7 least a sampled subset of objects allocated by the storage allocator.

1 31. The object sampling facility of claim 30, further comprising:  
2 an object sampler responsive to garbage collection events in the computational  
3 system, the object sampler referencing the sampled subset via the weak  
4 reference instances and maintaining object lifetime statistics based on  
5 the associated allocation-time information and then-current state of the  
6 sampled subset.

1 32. The object sampling facility of claim 30, further comprising:  
2 an object sampler referencing the sampled subset via the weak reference  
3 instances and maintaining object lifetime statistics based on the  
4 associated allocation-time information and sampled state of the  
5 sampled subset.

1 33. The object sampling facility of claim 32,  
2 wherein the storage allocator is responsive to the object lifetime statistics in its  
3 allocation decisions.

1 34. The object sampling facility of claim 30, further comprising:  
2 an object sampler referencing the sampled subset via the weak reference  
3 instances and maintaining object lifetime statistics based on the  
4 associated allocation-time information and sampled state of the  
5 sampled subset,

1 35. The object sampling facility of claim 34,

2 wherein a generational garbage collector is responsive to the object lifetime  
3 statistics in its promotion decisions.

1 36. The object sampling facility of claim 30, wherein the allocation-time  
2 information is indicative of one or more of:  
3 a time of object allocation;  
4 an allocation site; and  
5 an allocating thread.

1 37. The object sampling facility of claim 30, embodied as a computer  
2 program product.

1 38. A computer program product encoded in at least one computer readable  
2 medium, the computer program product comprising:  
3 at least one functional sequence for associating allocation-time information  
4 and an instance of a weak reference at least a sampled subset of objects  
5 allocated by a storage allocator; and  
6 at least one functional sequence for sampling the sampled subset using the  
7 weak reference instances and maintaining object lifetime statistics  
8 based on the associated allocation-time information and sampled state  
9 of the sampled subset.

1 39. A computer program product as recited in 38, embodied as a generational  
2 garbage collector and further comprising:  
3 at least one functional sequence for tenuring certain object instances in  
4 accordance with those of the object lifetime statistics corresponding  
5 thereto.

1 40. A computer program product as recited in 38, embodied as a generational  
2 run-time profiler.

1 41. A computer program product as recited in 38,

wherein the at least one computer readable medium is selected from the set of  
a disk, tape or other magnetic, optical, or electronic storage medium  
and a network, wireline, wireless or other communications medium.

42. An apparatus comprising:

means for associating allocation-time information with sampled instances of  
software objects;

means for referencing the sampled instances of software objects, the  
referencing means operable for both reachable and unreachable ones  
thereof;

means for updating lifetime predictions for categories of the software objects  
based on run-time access to states of corresponding ones of the  
sampled instances and associated allocation-time information therefor.

*add  
a<sup>8</sup>*